da\_410\_project2\_grahn

Jason Grahn

1/14/2019

## Download airpoll.txt. In this problem, we will only focus on the first 16 observations (cities). Read the data into R (as a data frame) and name the data as airpol.full. Then use the following code to “extract” the first 16 observations.

#import the dataset dataset as a better table.  
airpol.full <- read.csv("airpoll.txt", header = TRUE, sep = " ")  
  
#This is old bad code.  
#airpol.full <- read.table("http://www.stat.sc.edu/~hitchcock/airpoll.txt", header=T)  
  
#extract the first 16 observations  
airpol.data.sub <- head(airpol.full, 16)

## Display the subset data airpol.data.sub

#show that table  
airpol.data.sub

## City Rainfall Education Popden Nonwhite NOX SO2 Mortality  
## 1 akronOH 36 11.4 3243 8.8 15 59 921.9  
## 2 albanyNY 35 11.0 4281 3.5 10 39 997.9  
## 3 allenPA 44 9.8 4260 0.8 6 33 962.4  
## 4 atlantGA 47 11.1 3125 27.1 8 24 982.3  
## 5 baltimMD 43 9.6 6441 24.4 38 206 1071.0  
## 6 birmhmAL 53 10.2 3325 38.5 32 72 1030.0  
## 7 bostonMA 43 12.1 4679 3.5 32 62 934.7  
## 8 bridgeCT 45 10.6 2140 5.3 4 4 899.5  
## 9 bufaloNY 36 10.5 6582 8.1 12 37 1002.0  
## 10 cantonOH 36 10.7 4213 6.7 7 20 912.3  
## 11 chatagTN 52 9.6 2302 22.2 8 27 1018.0  
## 12 chicagIL 33 10.9 6122 16.3 63 278 1025.0  
## 13 cinnciOH 40 10.2 4101 13.0 26 146 970.5  
## 14 clevelOH 35 11.1 3042 14.7 21 64 986.0  
## 15 colombOH 37 11.9 4259 13.1 9 15 958.8  
## 16 dallasTX 35 11.8 1441 14.8 1 1 860.1

airpol.data.num.sub <- airpol.data.sub %>% select(Rainfall:Mortality)

## Use R to perform the following analysis on the subset data airpol.data.sub. Make sure you include clear headings, command lines, and relevant output/results.

### a) Calculate the sample covariance matrix and the sample correlation matrix.

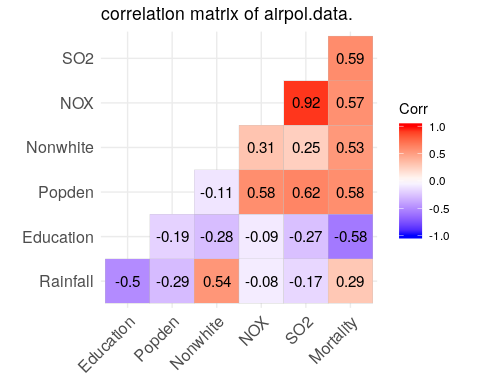
#sample covariance matrix  
S <- round(var(airpol.data.num.sub), 2)  
S

## Rainfall Education Popden Nonwhite NOX SO2  
## Rainfall 39.72 -2.46 -2766.77 34.61 -8.30 -84.69  
## Education -2.46 0.61 -224.44 -2.25 -1.14 -16.09  
## Popden -2766.77 -224.44 2229957.93 -1665.31 14294.73 71437.88  
## Nonwhite 34.61 -2.25 -1665.31 102.69 51.96 198.87  
## NOX -8.30 -1.14 14294.73 51.96 268.60 1169.95  
## SO2 -84.69 -16.09 71437.88 198.87 1169.95 5981.26  
## Mortality 101.49 -24.96 47577.01 294.06 513.19 2502.85  
## Mortality  
## Rainfall 101.49  
## Education -24.96  
## Popden 47577.01  
## Nonwhite 294.06  
## NOX 513.19  
## SO2 2502.85  
## Mortality 3030.05

#sample correlation matrix  
R <- round(cor(airpol.data.num.sub), 2)  
R

## Rainfall Education Popden Nonwhite NOX SO2 Mortality  
## Rainfall 1.00 -0.50 -0.29 0.54 -0.08 -0.17 0.29  
## Education -0.50 1.00 -0.19 -0.28 -0.09 -0.27 -0.58  
## Popden -0.29 -0.19 1.00 -0.11 0.58 0.62 0.58  
## Nonwhite 0.54 -0.28 -0.11 1.00 0.31 0.25 0.53  
## NOX -0.08 -0.09 0.58 0.31 1.00 0.92 0.57  
## SO2 -0.17 -0.27 0.62 0.25 0.92 1.00 0.59  
## Mortality 0.29 -0.58 0.58 0.53 0.57 0.59 1.00

#Identify which pairs of variables seem to be strongly associated  
#and describe the nature (strength and direction) of the relationship between these variable pairs.  
library(ggcorrplot)  
ggcorrplot(R,   
 type = "lower",  
 title = "correlation matrix of airpol.data.",  
 show.legend = TRUE,  
 digits = 2,  
 lab = TRUE)



From the plot we can see NOX and SO2 have the strongest correlation of the data, headed positive. the most negative is Education and Mortality. SO2 and Popden is also a semi-strong positive relationship. Everything else sits in a sort-of weak relationship.

### (b) Calculate the distance matrix for these observations (after scaling the variables by dividing each variable by its standard deviation).

#find standard deviation for variables  
std <- sapply(airpol.data.num.sub, sd)   
  
#divide each variable by its std to normalize the data  
airpol.data.sub.std <- sweep(x = airpol.data.num.sub, #sweep the airpol data  
 MARGIN = 2, #dunno what the MARGIN   
 STATS = std, #using the standard deviation  
 FUN = "/") #by division   
  
#find the distance matrix for the dataframe using all the default options  
dis <- dist(airpol.data.sub.std)  
  
dist2full <- function(ds) {   
 n <- attr(ds,"Size")   
 full <- matrix(0,n,n)   
 full[lower.tri(full)] <- ds   
 full + t(full)   
 }   
  
dis.matrix <- round(dist2full(dis), 2)  
dis.matrix

## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]  
## [1,] 0.00 1.76 2.80 2.84 5.14 4.81 2.09 2.21 2.92 1.34 4.15 5.00 2.40  
## [2,] 1.76 0.00 2.22 3.13 4.53 4.88 2.62 2.90 1.74 1.67 3.97 4.85 2.38  
## [3,] 2.80 2.22 0.00 3.23 4.51 4.56 3.42 2.18 2.45 2.03 2.98 5.69 2.40  
## [4,] 2.84 3.13 3.23 0.00 4.53 2.61 3.42 2.82 3.57 3.08 2.29 5.73 2.94  
## [5,] 5.14 4.53 4.51 4.53 0.00 3.62 5.05 5.87 3.74 5.11 4.40 3.14 3.00  
## [6,] 4.81 4.88 4.56 2.61 3.62 0.00 4.91 4.75 4.81 5.02 2.49 5.46 3.62  
## [7,] 2.09 2.62 3.42 3.42 5.05 4.91 0.00 3.25 3.21 2.72 4.77 4.63 2.98  
## [8,] 2.21 2.90 2.18 2.82 5.87 4.75 3.25 0.00 3.86 2.03 3.23 6.54 3.17  
## [9,] 2.92 1.74 2.45 3.57 3.74 4.81 3.21 3.86 0.00 2.32 4.25 4.56 2.57  
## [10,] 1.34 1.67 2.03 3.08 5.11 5.02 2.72 2.03 2.32 0.00 4.01 5.46 2.51  
## [11,] 4.15 3.97 2.98 2.29 4.40 2.49 4.77 3.23 4.25 4.01 0.00 6.37 3.29  
## [12,] 5.00 4.85 5.69 5.73 3.14 5.46 4.63 6.54 4.56 5.46 6.37 0.00 3.60  
## [13,] 2.40 2.38 2.40 2.94 3.00 3.62 2.98 3.17 2.57 2.51 3.29 3.60 0.00  
## [14,] 1.42 1.59 2.92 2.46 4.27 4.02 2.64 2.88 2.68 2.09 3.60 4.38 1.94  
## [15,] 1.41 1.71 3.16 2.50 5.13 4.69 2.12 2.87 2.57 1.87 4.24 5.25 2.99  
## [16,] 2.16 3.57 4.21 3.52 6.96 5.77 3.71 2.55 4.73 2.68 4.95 6.88 4.23  
## [,14] [,15] [,16]  
## [1,] 1.42 1.41 2.16  
## [2,] 1.59 1.71 3.57  
## [3,] 2.92 3.16 4.21  
## [4,] 2.46 2.50 3.52  
## [5,] 4.27 5.13 6.96  
## [6,] 4.02 4.69 5.77  
## [7,] 2.64 2.12 3.71  
## [8,] 2.88 2.87 2.55  
## [9,] 2.68 2.57 4.73  
## [10,] 2.09 1.87 2.68  
## [11,] 3.60 4.24 4.95  
## [12,] 4.38 5.25 6.88  
## [13,] 1.94 2.99 4.23  
## [14,] 0.00 1.74 3.05  
## [15,] 1.74 0.00 2.68  
## [16,] 3.05 2.68 0.00

## Describe some of the most similar pairs of cities and some of the most different pairs of cities, giving evidence from the distance matrix.

dist\_m <- as.matrix(dis)  
dist\_m

## 1 2 3 4 5 6 7 8  
## 1 0.000000 1.763021 2.795931 2.837504 5.135897 4.813668 2.088138 2.206094  
## 2 1.763021 0.000000 2.221206 3.130094 4.526527 4.884770 2.618902 2.898189  
## 3 2.795931 2.221206 0.000000 3.232179 4.511682 4.561887 3.418726 2.177101  
## 4 2.837504 3.130094 3.232179 0.000000 4.533080 2.605299 3.415994 2.821186  
## 5 5.135897 4.526527 4.511682 4.533080 0.000000 3.617197 5.053812 5.865775  
## 6 4.813668 4.884770 4.561887 2.605299 3.617197 0.000000 4.914560 4.748352  
## 7 2.088138 2.618902 3.418726 3.415994 5.053812 4.914560 0.000000 3.251476  
## 8 2.206094 2.898189 2.177101 2.821186 5.865775 4.748352 3.251476 0.000000  
## 9 2.924625 1.741600 2.449329 3.566846 3.743006 4.809554 3.211719 3.855911  
## 10 1.335913 1.668954 2.031965 3.084401 5.113896 5.023412 2.721293 2.032797  
## 11 4.154681 3.974604 2.982657 2.290831 4.404949 2.489499 4.772934 3.230345  
## 12 5.000224 4.845881 5.686235 5.728339 3.137265 5.464095 4.632264 6.539919  
## 13 2.396209 2.382967 2.401959 2.941879 3.000722 3.615936 2.976952 3.166743  
## 14 1.422266 1.589927 2.918292 2.455043 4.266018 4.015812 2.642049 2.880418  
## 15 1.407742 1.708825 3.157548 2.498664 5.125624 4.690884 2.116238 2.870437  
## 16 2.155150 3.565230 4.211207 3.516605 6.956464 5.773134 3.706677 2.550434  
## 9 10 11 12 13 14 15 16  
## 1 2.924625 1.335913 4.154681 5.000224 2.396209 1.422266 1.407742 2.155150  
## 2 1.741600 1.668954 3.974604 4.845881 2.382967 1.589927 1.708825 3.565230  
## 3 2.449329 2.031965 2.982657 5.686235 2.401959 2.918292 3.157548 4.211207  
## 4 3.566846 3.084401 2.290831 5.728339 2.941879 2.455043 2.498664 3.516605  
## 5 3.743006 5.113896 4.404949 3.137265 3.000722 4.266018 5.125624 6.956464  
## 6 4.809554 5.023412 2.489499 5.464095 3.615936 4.015812 4.690884 5.773134  
## 7 3.211719 2.721293 4.772934 4.632264 2.976952 2.642049 2.116238 3.706677  
## 8 3.855911 2.032797 3.230345 6.539919 3.166743 2.880418 2.870437 2.550434  
## 9 0.000000 2.323287 4.251377 4.561256 2.566487 2.676255 2.570231 4.730233  
## 10 2.323287 0.000000 4.011176 5.460695 2.513498 2.090329 1.870635 2.677896  
## 11 4.251377 4.011176 0.000000 6.366446 3.287103 3.595087 4.239763 4.953432  
## 12 4.561256 5.460695 6.366446 0.000000 3.599773 4.378435 5.249002 6.883273  
## 13 2.566487 2.513498 3.287103 3.599773 0.000000 1.942735 2.987918 4.226265  
## 14 2.676255 2.090329 3.595087 4.378435 1.942735 0.000000 1.735581 3.054556  
## 15 2.570231 1.870635 4.239763 5.249002 2.987918 1.735581 0.000000 2.681845  
## 16 4.730233 2.677896 4.953432 6.883273 4.226265 3.054556 2.681845 0.000000

dis

## 1 2 3 4 5 6 7 8  
## 2 1.763021   
## 3 2.795931 2.221206   
## 4 2.837504 3.130094 3.232179   
## 5 5.135897 4.526527 4.511682 4.533080   
## 6 4.813668 4.884770 4.561887 2.605299 3.617197   
## 7 2.088138 2.618902 3.418726 3.415994 5.053812 4.914560   
## 8 2.206094 2.898189 2.177101 2.821186 5.865775 4.748352 3.251476   
## 9 2.924625 1.741600 2.449329 3.566846 3.743006 4.809554 3.211719 3.855911  
## 10 1.335913 1.668954 2.031965 3.084401 5.113896 5.023412 2.721293 2.032797  
## 11 4.154681 3.974604 2.982657 2.290831 4.404949 2.489499 4.772934 3.230345  
## 12 5.000224 4.845881 5.686235 5.728339 3.137265 5.464095 4.632264 6.539919  
## 13 2.396209 2.382967 2.401959 2.941879 3.000722 3.615936 2.976952 3.166743  
## 14 1.422266 1.589927 2.918292 2.455043 4.266018 4.015812 2.642049 2.880418  
## 15 1.407742 1.708825 3.157548 2.498664 5.125624 4.690884 2.116238 2.870437  
## 16 2.155150 3.565230 4.211207 3.516605 6.956464 5.773134 3.706677 2.550434  
## 9 10 11 12 13 14 15  
## 2   
## 3   
## 4   
## 5   
## 6   
## 7   
## 8   
## 9   
## 10 2.323287   
## 11 4.251377 4.011176   
## 12 4.561256 5.460695 6.366446   
## 13 2.566487 2.513498 3.287103 3.599773   
## 14 2.676255 2.090329 3.595087 4.378435 1.942735   
## 15 2.570231 1.870635 4.239763 5.249002 2.987918 1.735581   
## 16 4.730233 2.677896 4.953432 6.883273 4.226265 3.054556 2.681845

### (c) Display a plot that will help assess whether this data set comes from a multivariate normal distribution. What is your conclusion based on the plot?

#Copy the chisplot function into R   
chisplot <- function(x) {   
 if (!is.matrix(x)) stop("x is not a matrix")   
 ### determine dimensions   
 n <- nrow(x)   
 p <- ncol(x)   
 xbar <- apply(x, 2, mean)   
 S <- var(x)   
 S <- solve(S)   
 index <- (1:n)/(n+1)  
 xcent <- t(t(x) - xbar)   
 di <- apply(xcent, 1, function(x,S) x %\*% S %\*% x,S)   
 quant <- qchisq(index,p)   
 plot(quant, sort(di),   
 ylab = "Ordered distances",   
 xlab = "Chi-square quantile",   
 lwd=2,  
 pch=1)  
 }  
  
chisplot(as.matrix(airpol.data.num.sub))

